Конкурсное задание

Проектирование и внедрение Системы класса АРМ для ОАО «Бакай Банк»

Версия 1.0

Дата: 03.03.2023

Бишкек

Содержание

1. Введение.

1.1. Назначение документа.

Документ описывает конкурсное задание (далее RFP) на проектирование и внедрение Системы класса АРМ (далее Система) для ОАО «Бакай Банк». Документ выступает в качестве исходных технических, функциональных и интеграционных требований к проектированию и внедрению Системы класса АРМ.

1.2. Определения, обозначения и сокращения.

Таблица 1. Обозначения и сокращения.

Термин, сокращение	**Определение**
АРМ	Application Performance management
АСУ	Автоматизированная система управления
ИА	Исполнительный аппарат
ИФ	Источник финансирования
ССМ	Система Сервисного Мониторинга
БД	База данных
Базовые значения (Baseline)	Уровни показателей, по отношению к которым ведется контроль и регулирование проекта
Инфопанель (Dashboard)	Элемент интерфейса системы с визуальным представлением данных, сгруппированных по смыслу на одном экране для более легкого визуального восприятия информации, чаще всего с элементами инфографики. Наглядно показывает ключевые показатели и помогает принимать решения
ИС	Информационная система
ИТ	Информационные технологии
ЛВС	Локальная вычислительная сеть
ОС	Операционная система
ПО	Программное обеспечение
Сетевой протокол	Набор правил и действий (очерёдности действий), позволяющий осуществлять соединение и обмен данными между двумя и более включёнными в сеть устройствами
Событие	Сообщение программного обеспечения или его части, указывающее на произошедшие действия
СУБД	Система управления базами данных

ТЗ	Техническое задание
Транзакция	Процесс выполнения логической операции, состоящей из одного или нескольких действий пользователя и ответной реакции на них со стороны системы
ЦОД	Центр обработки данных
AD	Active Directory – служба каталогов, разработанная компанией Microsoft для операционных систем семейства Windows
APM	Application Performance Monitoring – управление производительностью приложений– это технологии в области анализа и управления качеством работы ИТ-сервисов
API	Application programming interface – интерфейс программирования приложений
GUI	Graphical User Interface – графический пользовательский интерфейс
IP	Internet Protocol – протокол межсетевой адресации
TCP	Transmission Control Protocol – протокол управления передачей данных
ОРД	Организационно-распорядительные документы
КТС	Комплекс технических средств, на которых будет развёрнута Система

1.3. Контроль изменений

Версия	Дата	Автор	Состав изменений

1.4. Коммерческие требования

Участнику конкурсной процедуры необходимо предоставить Коммерческое Предложение (КП), которое должно быть оформлено в виде PDF файла с печатью и подписью, с форматом страниц А4 + excel файл с расчетом и включать в себя следующие разделы:

- Состав и стоимость лицензий
- Состав, стоимость и продолжительность работ по внедрению и первичной настройке Системы
- Состав и стоимость технического обслуживания Системы и лицензий на период 3 года
- Состав и стоимость обучения специалистов ОАО «Бакай Банк» в процессе внедрения для дальнейшего самостоятельного расширения системы при необходимости.

Необходимо указать суммы «Итого» по каждому из упомянутых разделов и затем указать общую сумму «Итого».

Для анализа и расчёта необходимого количества лицензий и стоимости работ каждому участнику конкурса будет направлена необходимая дополнительная информация об объекте мониторинга по отдельному запросу после подписания NDA.

Обобщённые данные об объекте мониторинга внедряемой Системы:

Количество digital-продуктов, которые должны являться объектами внедрения Системы	не более 10 штук
Общее количество пользователей продуктов	не более 1 млн.
Общее количество виртуальных CPU	не более 320
Общее количество оперативной памяти всех виртуальных серверов продуктов	не более 1000 Гб
Поток обрабатываемых данных	не более 200 Мбит/с
Количество серверов	125 ед.

Продолжительность работ по внедрению и первичной настройки указывается без учета времени на подготовку серверной инфраструктуры на стороне Заказчика.

2. Цели и задачи проекта

2.1. Цель проекта

Целью проекта является внедрение Системы для мониторинга и диагностики проблем ИТ-систем и сервисов ОАО «Бакай Банк», позволяющей в режиме реального времени получать данные о производительности и доступности сервисов, с которыми работают пользователи, и которая в свою очередь должна решать задачи:

- повышение доступности и качества сервисов, предоставляемых Заказчиком;
- сокращение времени реагирования специалистов на инциденты по сервисам;
- оптимизация работы службы поддержки, за счёт проактивного мониторинга сервисов и своевременное оповещение ИТ-специалистов о деградации сервисов;
- сокращение расходов на поиск технических проблем с ИТ-системами;
- сокращение времени и затрат на локализацию проблем и восстановление работоспособности сервисов;
- выявление узких мест в работе сервисов;
- избежание/значительное снижение ошибок при использовании ИТ-систем;
- получение объективной информации о работоспособности информационных систем с точки зрения конечных пользователей;
- предоставление информации о состоянии ИТ-систем и сервисов для менеджмента в режиме реального времени, включая готовые и настраиваемые инфопанели (dashboards) - систему раннего оповещения и предотвращения ошибок.
- мониторинг операций конечных пользователей;
- мониторинг обработки запросов оконечных устройств на серверной части;
- мониторинг обработки запросов оконечных устройств на сетевой части;
- мониторинг обработки запросов оконечных устройств на клиентской части;
- мониторинг ключевых операций приложений;
- мониторинг KPI в разрезе геолокации пользовательских устройств.
- сокращение времени на поиск проблем и восстановление работоспособности ИС за счет быстрой локализации источников проблем;
- возможность обнаружения «узких» мест инфраструктуры за счет детального анализа операций пользователей;
- возможность планирования расширения инфраструктуры ИТ, основываясь на текущей производительности систем;
- снижение нагрузки на первую линию технической поддержки Заказчика за счет проактивного выявления проблем;

- контроль качества релизов.

2.2. Задачи проекта

Основной задачей создания Системы является решение следующих проблем:

- превышение времени, отведенного на выявление причин сбоев;
- сложности в определении ответственных за устранение сбоев, особенно в вопросах, лежащих в пограничных зонах ответственности;
- отсутствие объективной информации о качестве работы ИС, в виду существования отчетности только по отдельным инфраструктурным компонентам;
- отсутствие информации об опыте конечных пользователей, работающих с ИС и как следствие, отсутствие информации о степени их удовлетворенности;
- отсутствие информирования о деградации сервиса по факту обращений пользователей;
- некорректное планирование мощностей и отсутствие необходимой информации для анализа влияния изменений на доступность и производительность ИС;
- отсутствие сквозного идентификатора сессии, позволяющего связать воедино действия между устройством пользователя и устройств серверной части;
- отсутствие отслеживания работы веб и мобильных приложений, действий пользователей в приложениях
- отсутствие информации о влиянии сбоев и ошибок в ИТ-системах на конечных пользователей/клиентов.

2.3. Требования к результатам работ

Результатом работ по внедрению должна быть настроенная и готовая к промышленной эксплуатации Система. Под настроенной и готовой к промышленной эксплуатации системой понимается развёрнутое программное обеспечение, готовое к работе после поставки и завершения работ, предусмотренных планом проекта. Исполнителем должны быть оказаны работы по внедрению Системы Заказчика согласно требованиям, указанным ниже.

2.4. Общие требования к Исполнителю

2.4.1. Исполнитель должен иметь опыт работы не менее 3 лет в сфере внедрения систем мониторинга и диагностики проблем ИТ-систем и сервисов (необходимо предоставить информацию о дате регистрации ЮЛ и подтвержденных проектах, реализованных за последние два года).

2.4.2. Исполнитель должен иметь подтвержденный опыт успешно завершенных проектов на

предлагаемом решении (предоставить список реализованных проектов, краткое описание реализованной функциональности по данному направлению).

3. Функциональные требования к Системе

3.1. Общие требования

3.1.1. Система должна автоматически выполнять все задачи, связанные с мониторингом хостов, приложений, инфраструктуры и реальных пользователей. Система информирует обо всех аспектах работы приложений, связанных со здоровьем и производительностью, отслеживая все запросы и время отклика по всем пользовательским транзакциям в режиме реального времени.

3.1.2. Система представляет полные возможности по мониторингу инфраструктуры, работы приложений с углублением до кода приложения, мониторингу контейнеров и микросервисов, анализу корневой причины, автоматическому анализу зависимостей при помощи средств искусственного интеллекта, анализу поведения пользователей, измерению пользовательского опыта, автообнаружению топологии сервиса.

3.1.3. Система должна объединять реальный мониторинг пользователей и данных из приложений, чтобы получить полный обзор каждой цифровой транзакции, понимание поведение клиентов и визуализацию влияния проблем.

3.1.4. Система должна быть независима от того, какие языки используются, архитектуры приложения или запуска приложения в облаке, локально или гибридно.

3.1.5. Система должна предоставлять полную видимость на уровне кода, отслеживать каждую транзакцию на всех уровнях, без пробелов и слепых зон, и переходить от клика пользователя к уровню кода или запроса базы данных.

3.1.6. Система должна предоставлять полную картину приложения, каждого пользователя, вплоть до уровня кода, трассировок запросов, метрик, sql запросов и т.д.

3.1.7. Система в режиме реального времени должна предоставлять ответы и действия в отношении первопричины, поведенческого анализа и влияния на бизнес.

3.1.8. Система должна позволять создавать дашборды с выводом на них инфраструктурных, бизнес и клиентских метрик.

3.1.9. Система должна в первую очередь решать задачи оперативного контроля состояния информационных систем, оповещения персонала эксплуатации о сбоях или деградации производительности контролируемых объектов, сбора и хранения событийной информации для последующего анализа.

3.1.10. Система должна иметь модульную масштабируемую архитектуру и включать в себя ряд функциональных подсистем, объединенных общими моделями данных и потоками информации о событиях и метриках производительности контролируемых объектов.

3.1.11. Система должна полностью находится в периметре ЦОДов Заказчика.

3.1.12. Система должна работать в полностью изолированном сетевом контуре, без доступа к

глобальной сети Интернет, за исключением доступа к обновлениям системы и получению другой служебной информации.

3.1.13. Система должна обновляться не реже чем раз в 3 месяца, как в части устранения уязвимостей/ошибок, так и части получения нового функционала

3.1.14. Система должна запускаться как на bare metal, так и на виртуализированных средах.

3.1.15. Система должна уметь горизонтально масштабироваться.

3.2. Требования к используемым технологиям и архитектуре

Предъявляются следующие требования к используемым технологиям и архитектуре:

3.2.1. Мониторинг должен осуществляется путём сбора данных с агентов, которые установлены на необходимых компонентах приложений;

3.2.2. Сервер обработки и отображения данных - должен представлять собой сервер, производящий углублённый анализ данных, получаемых от агентов и выводить данные в виде диаграмм или дашбордов;

3.2.3. Подключение к Системе пользователей и администраторов должно осуществляться через тонкий клиент (WEB-клиент);

3.2.4. Система должна обеспечивать сбор информации в одну базу с нескольких агентов мониторинга;

3.2.5. Система должна накапливать данные за определенный (настраиваемый) период времени;

3.2.6. Система должна иметь возможность глубокого анализа транзакций Java, .Net, Node JS, PHP, Python, Go, Ruby с помощью устанавливаемого агента;

3.2.7. В Системе должна быть возможность сбора инфраструктурных метрик с серверов (объектов мониторинга) как с помощью агентов, так и без них (используя стандартные метрики и экспортеры Prometheus).

3.2.8. Агент Системы должен быть максимально универсален и встраиваться в мобильные приложения и на web-сайты заказчика без существенных доработок приложений или web-сайтов.

3.2.9. Система не должно снижать стабильность и быстродействие мобильного приложения, web-сайта и абонентского терминала.

3.2.10. В Системе должна быть предусмотрена возможность расширения функционала силами заказчика при возникновении у Заказчика потребности в новых метриках или изменениях в системах

3.2.11. Система должна отображать события автоматически в онлайн режиме до каждого клиента

3.2.12. Система должна обработки большого количества метрик и объема анализируемой базы.

3.2.13. Система должна иметь открытый API для получения данных мониторинга. Возможность получения потока входящих данных (для всех типов обрабатываемой информации – трейсы, события, метрики и т.д.) из смежных систем через программный интерфейс системы.

3.2.14. Система должна иметь открытый API для предоставления данных мониторинга. Возможность предоставления через программный интерфейс системы сторонним системам следующих данных: первичных и сырых данных мониторинга (метрики, трейсы, события), выявленные проблемы/инциденты).

3.2.15. Система должна определять медленные запросы исполняемые на стороне БД без ручного вмешательства.

3.2.16. Система должна обладать функциональностью мониторинга Баз Данных для Oracle, Postgres, MSSQL, MySQL, MongoDB.

3.2.17. Система должна связывать производительность приложения с инфраструктурными метриками сервера или виртуальной машины, на котором это приложение работает.

3.2.18. Режим сбора данных в онлайн режиме – 24x7.

3.2.19. Система должна иметь настриваемую глубину хранения данных.

3.2.20. Возможность self-мониторинга системы.

3.2.21. Система должна поддерживать анализ трейсов приложений, инструментированных по стандарту OpenTelemetry.

3.2.22. Система должна поддерживать поиск по логам.

3.2.23. Система должна обеспечивать гибкую настройку времени хранения данных телеметрии (метрик, трейсов).

3.2.24. Система при развертывании должна иметь возможность установки и функционирования в изолированной от интернета среде.

3.2.25. Система должна поддерживать мониторинг как монолитных приложений, так и построенных на технологиях Cloud Native (OpenShift, Kubernetes, Docker).

3.2.26. Система должна поддерживать развертывание агентов мониторинга в Kubernetes с помощью helm chart или daemonset.

3.2.27. Агент Системы должен автоматически распознавать все контейнеры, СУБД, брокеры сообщений, веб-сервера, очереди сообщений и другие поддерживаемые компоненты, которые запущены на хосте/кластере.

3.2.28. Система должна собирать все события просмотров страниц пользователей и предоставлять их детализацию без сэмплирования.

3.2.29. Система должна автоматически обнаруживать и собирать JavaScript ошибки из подключенных веб-приложений.

3.2.30. Система должна автоматически коррелировать сессии пользователей с вызванными в ходе сессии транзакциями на бэкенде.

3.2.31. Система должна автоматически собирать метрики: время загрузки страниц, количество JS ошибок, количество затронутых пользователей JS ошибкой, количество просмотров страниц.

3.2.32. Система должна предоставлять возможность отслеживания скорости загрузки Single Page Application (SPA), а также времени перехода между экранами.

3.2.33. Система должна предоставлять возможность добавлять дополнительные метаданные к отслеживанию для обогащения предоставляемой информации по каждой сессии пользователя и их последующего анализа.

3.2.34. Система должна поддерживать мониторинг и сбор данных производительности, в том числе трассировку исходящих HTTP запросов, для нативных мобильных приложений (iOS, Android).

3.2.35. Система должна предоставлять данные по взаимодействию пользователя с мобильными и веб приложениями (HTTP запросы, количество активных сессий, количество ошибок, скорость загрузки и выполнения клиентских операций).

3.2.36. Система должна автоматически коррелировать транзакции, инициированные на мобильных устройствах с транзакциями на бэкенде.

3.3. Требования к функционалу работы системы

Предъявляются следующие требования к функционалу работы системы:

3.3.1. Система должна предусматривать возможность мониторинга всех пользователей, всех транзакций для выбранного ИТ-сервиса, при условии полного покрытии агентами серверов приложения данного ИТ-сервиса.

3.3.2. В системе должна быть обеспечена возможность отслеживания полного пути прохождения операций между всеми компонентами сервиса с мониторингом полного времени выполнения данных операций.

3.3.3. Система должна иметь автоматическую идентификацию часто используемых и медленных запросов приложения.

3.3.4. В Системе должна быть обеспечена возможность автоматического построения топологии компонентов приложений.

3.3.5. Система должна иметь возможность идентификации операций с плохой производительностью, т.е. медленно выполняющихся операций.

3.3.6. Система должна иметь механизм автоматического расчет базовой линии для метрик

3.3.7. Система должна собирать информацию о времени выполнения операций и количестве операций пользователей при обращении к сервису.

3.3.8. Система должна иметь возможность предоставления статистики о времени выполнения операций с разбивкой по API и уровням приложения.

3.3.9. Система должна иметь возможность отправки оповещений, в случае если параметры операции превышают заранее заданный уровень.

3.3.10. Система должна обеспечивать сбор 100% запросов, параметры вызова, как на стороне пользовательского интерфейса, так и на стороне серверов приложений.

3.3.11. Дополнительная нагрузка (overhead) на сервера приложений не должна превышать 5% от потребления CPU и оперативной памяти потребляемой приложением.

3.3.12. Система должна поддерживать мониторинг производительности приложений, написанных на языках программирования Python, .NET Core, .NET Framework, PHP, Node JS, Go, Java, Ruby, JavaScript, TypeScript, Swift, Object-C.

3.3.13. Система должна поддерживать трейсинг приложений: Python, .NET Core, .NET Framework, PHP, Node JS, Go, Java, Ruby, JavaScript, TypeScript, Dart, Swift, Object-C.

3.3.14. Система должна поддерживать сбор метрик с СУБД: Oracle, MySQL, MariaDB, MongoDB, Elasticsearch, Postgresql, Redis, Cassandra, MS SQL.

3.3.15. Система должна поддерживать сбор метрик с веб серверов и серверов приложений, брокеров сообщений: NGINX, Apache, Tomcat, Kafka, JBoss, Microsoft IIS, Websphere, HAProxy.

3.3.16. Система должна поддерживать сбор метрик с ОС: Linux, Windows.

3.3.17. Система должна поддерживать сбор метрик с очередей сообщений: RabbitMQ, ActiveMQ, Kaffka.

3.3.18. Система должна поддерживать сбор метрик с S3 совместимого хранилища.

3.3.19. Для поддерживаемых языков программирования должна поддерживаться их автоматическая инструментация, то есть получение метрик производительности и данных о транзакциях без необходимости модификации исходного кода приложений или с минимальными модификациями (такими как подключение библиотеки, вызов библиотеки и тд).

3.3.20. Система должна поддерживать автоматическое подключение агентов/библиотек к приложениям, работающим в Kubernetes, для следующих языков разработки: Java, Node JS, Python. Автоматическое подключение агентов к приложениям подразумевает отсутствие необходимости модификации приложений для сбора трейсов и метрик приложений.

3.3.21. Система должна предоставлять возможность автоматической визуализации взаимосвязей приложений и компонентов приложений на основе анализа потока данных, без необходимости ручной настройки визуализации.

3.3.22. Система должна предоставлять возможность анализа запросов из приложений к базам данных.

3.3.23. Система должна автоматически выстраивать карту инфраструктуры, указывать на проблемные места в инфраструктуре.

3.3.24. Система должна предоставлять возможность добавления информации о релизе (дата и время выпуска релиза, версия релиза) и отображать эти данные на всех графиках производительности системы.

3.3.25. Система должна собирать все транзакции всех отслеживаемых приложений без сэмплирования и предоставлять по ним данные (стэк трейс ошибки, детали трейса, количество вложенных вызовов, спаны и время их исполнения, дерево вызовов, таймлайн трейса, информацию о инфраструктуре).

3.3.26. Система должна иметь возможность идентификации транзакций по значениям, передаваемым в HTTP запросах (захват параметров).

3.3.27. Система должна иметь функционал проверки доступности веб приложений, с возможностью фиксирования последовательности действий пользователя и воспроизведения этой последовательности действий.

3.3.28. Система должна автоматически предоставлять следующие данные о Kubernetes: namespace, pods, nodes и тд.

3.3.29. Система должна автоматически собирать следующие метрики с Kubernetes кластера: CPU Requsets, CPU limits, Memory Limits, Memory Requests, количество запущенных Pods.

3.3.30. Система должна иметь встроенные и работающие “из коробки” правила оповещений, по которым пользователи системы будут получать оповещения в случаях деградации производительности и автоматически подсказывать пользователям о возможной причине деградации работы приложения.

3.3.31. Система должна иметь возможность создания пользовательских правил оповещений, по которым пользователи системы будут получать оповещения в случаях деградации производительности.

3.3.32. Система должна иметь возможность отправлять созданные оповещения по WebHook или иметь готовую интеграцию с AlertManager.

3.3.33. Система должна предоставлять возможность фильтрации инцидентов по сущностям инфраструктуры и другим объектам мониторинга.

3.3.34. API системы должно позволять создавать и гибко настраивать доступ пользователей к различным функциям продукта.

3.4. Требования к отчётам, создаваемым Системой

3.4.1. Система должна предоставлять готовые дашборды с основными метриками для всех поддерживаемых технологий/языков программирования/сервисов.

3.4.2. Система должна иметь возможность выводить на дашборды метрики производительности, доступности и успешности по отдельным клиентским операциям.

3.4.3. Должна быть возможность создания пользовательских показателей и метрик и вывода их на дашборды и в отчеты.

3.5. Требования к характеристикам взаимосвязей создаваемой Системы со смежными системами

3.5.1. Для обеспечения аутентификации пользователей в Системе с использованием учетной записи необходимо обеспечить интеграцию с корпоративным SSO на базе протокола SAML.

3.5.2. Должно быть организовано взаимодействие Системы с корпоративной службой электронной почты Заказчика.

3.6. Требования по диагностированию Системы

3.6.1. Система должна предоставлять инструменты диагностирования основных своих метрик производительности и работы (самодиагностика).

3.6.2. Компоненты Системы должны иметь удобный интерфейс для возможности просмотра метрик самодиагностики.

3.7. Требования к хранению данных

3.7.1. Система должна предоставлять возможность накопления данных за определенный (настраиваемый) период.

3.7.2. Детализация информации и максимальное время хранения статистических данных должны определяться свободным дисковым пространством серверов Системы и производительностью БД. При необходимости, Заказчик должен иметь возможность настроить максимальное время хранения самостоятельно. По истечении заданного периода времени хранения, статистические данные должны удаляться в автоматическом режиме.

3.8. Требования к отказоустойчивости Системы

Система должна обеспечивать мониторинг всех пользователей, всех транзакций и всех приложений в режиме 24 часа в сутки, 7 дней в неделю, 365 дней в году и обеспечивает восстановление после устранения сбоев, отказов и аварий на программных, аппаратных и сетевых участках, а также обеспечивает сохранность данных при сбоях в электропитании оборудования. Также система должна обеспечивает возможность резервного копирования и восстановления конфигурации системы.

4. Требования к проведению работ по внедрению системы

В рамках внедрения должны быть проведены следующие работы по обследованию, установке и настройке Системы:

1. Проведение пилотного использования на органичном наборе систем;
2. Обследование инфраструктуры заказчика, подготовка требований и плана внедрения;
3. Установка серверного ПО (отказоустойчивая схема);
4. Сопровождение и консультации по установке агентов на целевые хосты;
5. Настройка глобальных параметров, глубины хранения данных;
6. Разделение окружения на сегменты, для предоставления доступа;
7. Создание пользователей, ролей, групп;
8. Настройка глобальных правил тегирования хостов, сервисов, приложений;
9. Настройка глобальных порогов для отслеживания здоровья инфраструктурных показателей, процессных метрик производительности;
10. Подключение к e-mail;
11. Интеграция с Active Directory/SAML;
12. Согласование и перечня оповещений и их настройка;
13. Настройка интеграции со смежными системами

Установку агентов на целевые хосты сервисов осуществляет Заказчик. Исполнитель сопровождает работы на стороне Системы и осуществляет консультации специалистов Заказчика.

Необходимые работы по настройке интеграций со смежными системами на стороне соответствующих внешних систем осуществляет Заказчик. Исполнитель выполняет работы по интеграции только на стороне ПО Системы.

После завершения установки Системы, Исполнитель проводит работы по настройке системы:

14. Настройка порогов производительности по ключевым запросам.
15. Настройка компонентов мониторинга реальных пользователей, выделение приложений и их настройка, настройка правил захвата имен пользователей.
16. Создание и настройка панелей «дашборд» (до 20 дашбордов).
17. Сбор требований и настройка автоматических уведомлений в системе мониторинга;
18. Демонстрация результатов.

5. Требования к сопровождению и технической поддержке

В рамках сопровождения и технической поддержки должны быть соблюдены следующие условия:

1. Обработка запросов осуществляется на русском языке по электронной почте, по будням с 09:00 до 18:00 по UTC +6 (Бишкек).
2. На каждый принятый запрос, генерируется и высылается заказчику письмо-подтверждение, содержащее номер запроса и дату принятия запроса.

Для инцидентов применяются следующие уровни обслуживания:

Инцидент	Приоритет	Максимальное время ответа (первой реакции)
Неработоспособность компонентов системы	Обычный	4 часа
	*Критический	1 час
Неработоспособность приложений и инфраструктуры заказчика, вызванная нештатной работой компонентов системы или несовместимостью программного обеспечения заказчика и компонентов системы	Обычный	4 часа
	Критический	1 час

**Запрос категории Инцидент получает Критический приоритет, если возникла проблема с продакшен-средой инфраструктуры заказчика.*

Для консультаций применяется следующие параметры уровня обслуживания:

Консультация	Приоритет	Максимальное время ответа (первой реакции)
Консультация по часто возникающим вопросам (в том числе по общим вопросам «Как это работает?» и «Что это такое?») о компонентах и функциональных возможностях системы	Обычный	8 часов
Консультация и оказание помощи при настройке компонентов системы	Обычный	8 часов

Общие рекомендации по архитектуре и размещению компонентов системы	Обычный	16 часов
Консультация по архитектурным решениям с учетом сценария заказчика по использованию системы	Обычный	16 часов
Консультация по интеграции сторонних решений с системой	Обычный	16 часов